



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|--|-------------|------------------------|---------------------|------------------|
| 09/754,785 | 01/04/2001 | Pierre-Alain Darlet | 40101/06901 | 3238 |
| 30636 7590 07/17/2008 FAY KAPLUN & MARCIN, LLP 150 BROADWAY, SUITE 702 NEW YORK, NY 10038 | | | | |
| EXAMINER KISS, ERIC B | | | | |
| ART UNIT 2192 | | PAPER NUMBER | | |
| MAIL DATE 07/17/2008 | | DELIVERY MODE PAPER | | |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

09/754,785

Applicant(s)

DARLET, PIERRE-ALAIN

Examiner

Eric B. Kiss

Art Unit

2192

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 29 April 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-15 and 40-60 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-15 and 40-60 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SF/IC)
- Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
- Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(c), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(c) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on April 29, 2008, has been entered. Claims 1-15 and 40-60 are pending.

Response to Arguments

3. Applicant's arguments filed April 29, 2008, have been fully considered but they are not persuasive.

Levine discloses instructions operable to reorder components of the software module into a predetermined order (i.e., an order without backward references; see "Creating libraries" on pp. 5-6, and in particular, the discussion of using tsort and lorder to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references), the order being based on the type, i.e., objects that are referenced.

Claim Rejections - 35 USC § 102

4. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

5. Claims 1-15, 40, 41, and 43-60 are rejected under 35 U.S.C. 102(b) as being anticipated by John Levine, "Linkers and Loaders, chapter 6," June 1999 [online] accessed 08/15/2005,

Retrieved from Internet <URL: <http://www.iecc.com/linker/linker06.txt>>, 9 pages (hereinafter *Levine*).

As per claim 1, *Levine* discloses receiving a software module, the software module including references to locations within the software module, at least some of the references being backward references; and reordering components of the software module into a predetermined order based on a type (*i.e.* objects being referenced) of the components to remove at least some of the backward references (see “Creating libraries” on pp. 5-6, and in particular, the discussion of using *tsort* and *lorder* to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references), wherein the components include at least one of a header, a section, and a table (see p. 2 (Libraries consist of an archive header, followed by alternating file headers and object files)).

As per claim 2, *Levine* further discloses adjusting at least one of the references in the software module to reflect the reordering of the components of the software module, so that the at least one of the references remains a reference to the same component, by to the component’s new, reordered location, the new, reordered location coming after the at least one reference in the software module (see “Creating libraries” on pp. 5-6 and “Library formats” on pp. 1-5).

As per claims 3 and 4, *Levine* further discloses the software module including a symbol table, the symbol table including no backward references after the reordering and adjusting steps (see “Creating libraries” on pp. 5-6, and in particular, the discussion of using *tsort* and *lorder* to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references). Further, *Levine* discloses that under

certain circumstances, lorder and tsort won't be able to come up with a total order for the files, resulting in backward references remaining (see "Exercises" on p. 8).

As per claims 5-8, *Levine* further discloses the use of relocatable ELF object files, which include sections grouped into segments (see "Library formats" on pp. 1-5). *Levine* further discloses the software module including a symbol table, the symbol table including no backward references after the reordering and adjusting steps (see "Creating libraries" on pp. 5-6, and in particular, the discussion of using tsort and lorder to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references). Further, *Levine* discloses that under certain circumstances, lorder and tsort won't be able to come up with a total order for the files, resulting in backward references remaining (see "Exercises" on p. 8).

As per claim 9, *Levine* discloses a reorder module configured to receive a software module including references to locations within the software module, at least some of the references being backward references, the reorder module configured to reorder components of the software module into a predetermined order based on a type of the components (*i.e.* objects being referenced) and remove at least some of the backward references (see "Creating libraries" on pp. 5-6, and in particular, the discussion of using tsort and lorder to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references), the components including at least one of a header, a section, and a table (see p. 2 (Libraries consist of an archive header, followed by alternating file headers and object files)). The use of a processor and memory is inherent in realizing the functionality of *Levine*.

As per claims 10, *Levine* further discloses adjusting at least one of the references in the software module to reflect the reordering of the components of the software module, so that the at least one of the references remains a reference to the same component, by to the component's new, reordered location, the new, reordered location coming after the at least one reference in the software module (see "Creating libraries" on pp. 5-6 and "Library formats" on pp. 1-5).

As per claims 11 and 12, *Levine* further discloses the software module including a symbol table, the symbol table including no backward references after the reordering and adjusting steps (see "Creating libraries" on pp. 5-6, and in particular, the discussion of using *tsort* and *lorder* to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references). Further, *Levine* discloses that under certain circumstances, *lorder* and *tsort* won't be able to come up with a total order for the files, resulting in backward references remaining (see "Exercises" on p. 8).

As per claims 13-15, *Levine* further discloses the use of relocatable ELF object files, which include sections grouped into segments (see "Library formats" on pp. 1-5). *Levine* further discloses the software module including a symbol table, the symbol table including no backward references after the reordering and adjusting steps (see "Creating libraries" on pp. 5-6, and in particular, the discussion of using *tsort* and *lorder* to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references). Further, *Levine* discloses that under certain circumstances, *lorder* and *tsort* won't be able to come up with a total order for the files, resulting in backward references remaining (see "Exercises" on p. 8).

As per claims 40 and 41, *Levine* further discloses linking the reordered module after the reordering (see “Creating libraries” on pp. 5-6, and in particular, the discussion of using *tsort* and *lorder* to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references).

As per claims 43-46, *Levine* further discloses the use of relocatable ELF object files, which include sections grouped into segments (see “Library formats” on pp. 1-5).

As per claims 47-54, *Levine* further discloses the reference pointing to/into a section or module before and after reordering (see “Creating libraries” on pp. 5-6 and “Library formats” on pp. 1-5).

As per claim 55, *Levine* discloses receiving a software module, the software module including components arranged in a first order, a first one of the components including a reference to a location in a second one of the components, the second one of the components preceding the first one of the components in the first order; and arranging the components into a predetermined second order so that the second one of the components is subsequent to the first one of the components in the second order, wherein the arrangement is based on a type (*i.e.* objects being referenced) of the first and second ones of the components (see “Creating libraries” on pp. 5-6, and in particular, the discussion of using *tsort* and *lorder* to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references), wherein the components include at least one of a header, a section, and a table (see p. 2 (Libraries consist of an archive header, followed by alternating file headers and object files)).

As per claims 56 and 57, *Levine* further discloses linking the reordered module after the reordering (see “Creating libraries” on pp. 5-6, and in particular, the discussion of using *tsort* and *lorder* to arrange object files within an archive library in proper dependency order to allow a single sequential linker pass to resolve all undefined references).

As per claim 58-60, *Levine* further discloses the use of relocatable ELF object files, which include sections grouped into segments (see “Library formats” on pp. 1-5).

Claim Rejections - 35 USC § 103

6. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.
7. Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Levine* as applied to claim 1 above, and further in view of U.S. Patent No. 6,185,733 to Breslau et al.

As per claim 42, *Levine* discloses such a method but fails to expressly disclose transferring the reordered module to a different computer system and linking the module on the different computer system. However, *Breslau et al.* teaches the use of remote object libraries distributed prior to linking (see, for example, col. 4, lines 11-20). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to such use of a different computer for linking. One would be motivated to do so, for example, to facilitate distributed software development efforts or reduce the physical storage requirements for object files (see, for example, col. 2, lines 4-25).

Conclusion

8. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The

Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eric B. Kiss/
Eric B. Kiss
Primary Examiner, Art Unit 2192